Int J Comput Vis
DOI 10.1007/s11263-010-0349-3

Plane-Based Calibration for Linear Cameras

Jamil Draréni · Sébastien Roy · Peter Sturm

Received: 1 May 2009 / Accepted: 29 April 2010
© Springer Science+Business Media, LLC 2010

Abstract Linear or 1D cameras are used in several areas such as industrial inspection and satellite imagery. Since 1D cameras consist of a linear sensor, a motion (usually perpendicular to the sensor orientation) is performed in order to acquire a full image. In this paper, we present a novel linear method to estimate the intrinsic and extrinsic parameters of a 1D camera using a planar object. As opposed to traditional calibration scheme based on 3D-2D correspondences of landmarks, our method uses homographies induced by the images of a planar object. The proposed algorithm is linear, simple and produces good results as shown by our experiments.

Keywords Pushbroom camera · Planar calibration · Linear sensor

1 Introduction

Pushbroom cameras or linear scanners are a one-dimensional imaging devices. They are preferred over conventional 2D cameras when it comes to *scan* a static scene like airborne landscapes and urban scapes reconstruction (Hirschmüller and Hirzinger 2005). This choice is motivated by the need for a higher frame rate and a better resolution. At the time of writing, existing pushbroom cameras embed sensors up to 8192 pixels and delivers 1D images at a stunning frame-rate of 140 KHz (Basler 2009).

If the acquired images are meant for a 3D euclidean reconstruction or metrology purposes (Ishiguro and Tsuji 1992; Shum and Szeliski 1999), a camera calibration is necessary. As detailed in Sect. 2 linear cameras have a specific model thus, standard 2D camera calibration methods can no longer be used to recover internal parameters.

Classical calibration methods use mappings of 3D feature points on a calibration rig and their projections on the image to infer the internal parameters of a camera (Tsai 1986; Faugeras 1987). These methods are not very flexible because they use a specially designed calibration rig and often, features are manually selected.

In the last decade, new plane-based calibration methods have been introduced (Sturm and Maybank 1999; Zhang 1999). They enjoyed a growing popularity in the computer vision community due to their stability and their higher ease of use. In fact, the calibration can be done with an off-the-shelf planar object and a printed checkerboard.

Despite the several improvements that plane-based calibration methods went through (Gurdjos and Payrissat 2001; Wang et al. 2006; Ricolfe Viala and Sanchez Salmeron 2005), none of these works tackled the calibration of linear cameras. In fact, the predominant method for 1D camera calibration was proposed by Hartley et al. (Rajiv Gupta 1997; Gupta and Hartley 1995) and supposes a mapping between 3D landmarks and their projections in the image.

In this paper, we present a novel method to fully calibrate a pushbroom camera using a planar object. Here, the considered camera model is the translational pushbroom camera.

J. Draréni (✉) · S. Roy
Département d'Informatique et Recherche Opérationnelle, Université de Montréal, CP 6128 Succ. Centre-Ville, Montréal, Québec H3C 3J7, Canada
e-mail: drarenij@iro.umontreal.ca

S. Roy
e-mail: roys@iro.umontreal.ca

P. Sturm
INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 Montbonnot, France
e-mail: Peter.Sturm@inrialpes.fr

Published online: 15 May 2010

Our method is linear, fast and simple to implement. To the best of our knowledge, the presented plane-based calibration is the first of its kind.

For the rest of the paper, the terms 1d camera, linear camera and pushbroom camera will be used equally.

The remaining of the paper is organized as follows, in Sect. 2, the linear camera model is described. Section 3, presents the mathematical derivation and the algorithm of the plane-based calibration for linear cameras. The experiments and results are reported in Sect. 5 and we finally summarize our conclusion in Sect. 6.

2 Camera Model

Although the motivation behind our work was to calibrate a flatbed scanner using a pushbroom model, the presented algorithm along with the mathematical derivations still hold for any linear camera provided that the sensor undergoes a linear constant motion orthogonal to its orientation.

In general, a 1D camera consists of a linear array of sensors (such as CCD) recording an image projected by an optical system. A displacement of the sensor (usually orthogonal to the sensor) is required. We make the same reasonable assumption as in Rajiv Gupta (1997) regarding the sensor motion. We assume its velocity constant. We set up the local camera coordinate system as depicted in the Fig. 1. Let the point $(u, v, 1)^\mathsf{T}$ be the projection of the 3D point $(X, Y, Z)^\mathsf{T}$ in the camera image plane. The perspective projection of the coordinate u along the sensor can be modelled with a 2×3 projection matrix P:

$$\begin{pmatrix} u \\ 1 \end{pmatrix} \sim \underbrace{\begin{pmatrix} f & u_0 & 0 \\ 0 & 1 & 0 \end{pmatrix}}_{\mathsf{P}} \begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} \tag{1}$$

The parameters f and u_0 are respectively the focal length and the principal point of the linear sensor. As the sensor sweeps the scene, a 2D image is formed by stacking the 1D images obtained through the successive camera positions. Since the speed of the camera is assumed constant, the v coordinates is related to Y by a scaling factor s that depends on the speed of the sensor:

$$v = sY \tag{2}$$

If we combine (1) and (2) in a single matrix, the complete projection of a 3D point $(X, Y, Z)^\mathsf{T}$ is expressed as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fX + u_0 Z \\ sYZ \\ Z \end{pmatrix} \sim \underbrace{\begin{pmatrix} f & 0 & u_0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathsf{K}} \begin{pmatrix} X \\ YZ \\ Z \end{pmatrix} \tag{3}$$

where K represents the sought intrinsic camera matrix. We can see from the above equation that the perspective coordinate u depends solely on X and its depth Z, whereas v the orthographic coordinate is directly related to Y and the scaling factor s. One can also observe the non-linearity of the projection equation in the 3D coordinates due to the YZ term. This is not surprising, since the projection is non-central. This precludes the use of a pinhole-based camera calibration.

3 Calibration with a Planar Grid

Let us consider a point $(a, b, 0)^\mathsf{T}$ on the grid. It is mapped into the camera's coordinate system as $(X, Y, Z)^\mathsf{T}$ by a rigid transform:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathsf{R} \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} + \mathbf{t} \tag{4}$$

where R is a 3×3 rotation matrix and $\mathbf{t}$ a translation vector. Notice that, since the considered point lies on the grid, its third coordinate is null. Hence, the entries of the third column of R are zeroed and (4) in homogeneous coordinates simplifies as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = (\mathbf{R}_1 \quad \mathbf{R}_2 \quad \mathbf{t}) \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} = \begin{pmatrix} ar_{11} + br_{12} + t_1 \\ ar_{21} + br_{22} + t_2 \\ ar_{31} + br_{32} + t_3 \end{pmatrix} \tag{5}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are the two first columns of R. As stated before, the non-central nature of the camera makes it impossible to establish a linear mapping between points on the grid and their images on the camera plane. For instance, $(u, v, 1)^\mathsf{T}$ is expressed from (3) and (5) as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{K} \begin{pmatrix} X \\ YZ \\ Z \end{pmatrix} = \mathsf{K} \begin{pmatrix} ar_{11} + br_{12} + t_1 \\ a(r_{21}t_3 + r_{31}t_2) + b(r_{22}t_3 + r_{32}t_2) + t_2t_3 + a^2 r_{21}r_{31} + b^2 r_{22}r_{32} + ab(r_{21}r_{32} + r_{22}r_{31}) \\ ar_{31} + br_{32} + t_3 \end{pmatrix} \tag{6}$$

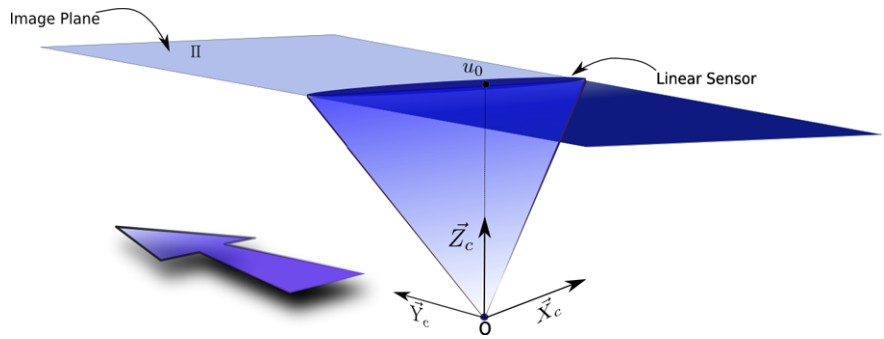

Fig. 1 A typical linear camera. A sensor, linear along the X-axis, undergoes motion along the Y-axis

An approach to circumvent this problem is to express the points in a higher dimensional space via the so-called "lifted" coordinates. In our case, the point $(a, b, 1)^\mathsf{T}$ "lifts" (according to their Veronese mapping) to $({}^\mathsf{T}a, b, 1, a^2, b^2, ab)$. Thus, (6) becomes:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{K} \cdot \underbrace{\begin{pmatrix} r_{11} & r_{12} & t_1 & 0 & 0 & 0 \\ r_{21}t_3 + r_{31}t_2 & r_{22}t_3 + r_{32}t_2 & t_2t_3 & r_{21}r_{31} & r_{22}r_{32} & r_{21}r_{32} + r_{22}r_{31} \\ r_{31} & r_{32} & t_3 & 0 & 0 & 0 \end{pmatrix}}_{\mathsf{T}} \begin{pmatrix} a \\ b \\ 1 \\ a^2 \\ b^2 \\ ab \end{pmatrix} \tag{7}$$

which represents the complete projection equation of a point on the grid expressed in its lifted coordinates. The homography $\mathsf{H} \sim \mathsf{KT}$, that maps points on the grid and their images in the camera, has 6 zeroed entries. The remaining 12 non-zero entries can be estimated up to a scale factor using 6 or more point matches as explained in the next subsection.

3.1 Estimate the Homography

We recall from (7) that the mapping between grid points and image points is represented by the homography H as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \mathsf{H} \cdot \begin{pmatrix} a \\ b \\ 1 \\ a^2 \\ b^2 \\ ab \end{pmatrix} \tag{8}$$

If we multiply both hands of the above equation by

$$\left[\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \right]_\times ,$$

the cross product skew matrix, we get a homogeneous equation system that upon simplifications yields the following linear and homogeneous equation system in the entries of H:

$$\begin{pmatrix} 0 & 0 & 0 & a & b & 1 & a^2 & b^2 & ab & -av & -bv & -v \\ a & b & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -au & -bu & -u \\ -av & -bv & -v & au & bu & u & a^2u & b^2 & abu & 0 & 0 & 0 \end{pmatrix} \mathbf{h} = \mathbf{0} \tag{9}$$

where $\mathbf{h}^\mathsf{T} = (\mathbf{h}_{11}, \mathbf{h}_{12}, \mathbf{h}_{13}, \mathbf{h}_{21}, \mathbf{h}_{22}, \mathbf{h}_{23}, \mathbf{h}_{24}, \mathbf{h}_{25}, \mathbf{h}_{26}, \mathbf{h}_{31}, \mathbf{h}_{32}, \mathbf{h}_{33})$ is the vector that contains the non-zero entries of H.

It is easy to see that only two equations are linearly independent. For instance, the third row can be obtained by adding the first and the second row, scaled respectively by u and v. Thus, given at least 6 matches between grid points and their images, H can be solved using 2 equations from the system (9) per match.

3.2 Extracting the Principal Point and the Focal Length

We shall now show how the camera's internal parameters are extracted from the homographies computed in the previous subsection. Let us recall the explicit form of the homography H:

$$\mathsf{H} = \lambda \begin{pmatrix} fr_{11} + u_0 r_{31} & fr_{12} + u_0 r_{32} & ft_1 + u_0 t_3 & 0 & 0 & 0 \\ s(r_{21}t_3 + r_{31}t_2) & s(r_{22}t_3 + r_{32}t_2) & st_2t_3 & sr_{21}r_{31} & sr_{22}r_{32} & s(r_{21}r_{32} + r_{22}r_{31}) \\ r_{31} & r_{32} & t_3 & 0 & 0 & 0 \end{pmatrix} \tag{10}$$

The scalar λ is added because the homography H can only be retrieved up to a scale factor. One can notice that $\bar{\mathsf{R}}$, the two first rotation's columns can be expressed as:

$$\bar{\mathsf{R}} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \\ r_{31} & r_{32} \end{pmatrix} = \begin{pmatrix} \frac{h_{11} - u_0 h_{31}}{\lambda f} & \frac{h_{12} - u_0 h_{32}}{\lambda f} \\ \frac{h_{24}}{s h_{31}} & \frac{h_{25}}{s h_{32}} \\ \frac{h_{31}}{\lambda} & \frac{h_{32}}{\lambda} \end{pmatrix} \tag{11}$$

From the above equation, $\bar{\mathsf{R}}$ can be expressed as a product of two matrices (up to a scale factor) L that depends on internal parameters and M:

$$\begin{aligned} \mathsf{L} &= \begin{pmatrix} s & 0 & -su_0 \\ 0 & \lambda f & 0 \\ 0 & 0 & sf \end{pmatrix}, \\ \mathsf{M} &= \begin{pmatrix} H_{11} & H_{12} \\ H_{24}/H_{31} & H_{25}/H_{32} \\ H_{31} & H_{32} \end{pmatrix} \end{aligned} \tag{12}$$

The product of $\bar{\mathsf{R}}$ with its transpose is a 2×2 identity matrix due to the orthogonality of its columns. Thus, we have:

$$\bar{\mathsf{R}}^{\mathsf{T}} \bar{\mathsf{R}} = \mathtt{I}_{2\times 2} \sim \mathsf{M}^{\mathsf{T}} \mathsf{L}^{\mathsf{T}} \mathsf{L} \mathsf{M} \tag{13}$$

The matrix L is related to the above calibration matrix K, with the notable fact that it also includes the scalar λ. Note that λ will be different for each scan, as opposed to the 3 intrinsic parameters f, s and u_0 which remain the same.

Let us define the matrix X as:

$$\begin{aligned} \mathsf{X} = \mathsf{L}^{\mathsf{T}} \mathsf{L} &= \begin{pmatrix} s^2 & 0 & -s^2 u_0 \\ 0 & \lambda^2 f^2 & 0 \\ -s^2 u_0 & 0 & s^2(u_0^2 + f^2) \end{pmatrix} \\ &= \begin{pmatrix} v_1 & 0 & v_2 \\ 0 & v_4 & 0 \\ v_2 & 0 & v_3 \end{pmatrix} \end{aligned} \tag{14}$$

where the intermediate variables v_1, v_2, v_3, v_4 were introduced for ease of notation. The equation (13) gives 2 constraints on X that can be written as:

$$\left.\begin{aligned} (\mathsf{M}^{\mathsf{T}} \mathsf{X} \mathsf{M})_{12} &= 0 \\ (\mathsf{M}^{\mathsf{T}} \mathsf{X} \mathsf{M})_{11} - (\mathsf{M}^{\mathsf{T}} \mathsf{X} \mathsf{M})_{22} &= 0 \end{aligned}\right\} \tag{15}$$

which in turn can be expressed in terms of the intermediate variables $v_{1,2,3,4}$ as:

$$\begin{pmatrix} m_{11}m_{12} & m_{11}m_{32} + m_{12}m_{31} & m_{31}m_{32} & m_{21}m_{22} \\ m_{11}^2 - m_{12}^2 & 2(m_{11}m_{31} - m_{12}m_{32}) & m_{31}^2 - m_{32}^2 & m_{21}^2 - m_{22}^2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{16}$$

With at least two different scan of a grid, the $v_{1,2,3,4}$ can be computed up to a scaling factor. Bare in mind that v_4 is different at each scan because of the homography scaling factor λ. Once the $v_{1,2,3,4}$ computed, the principal point and the focal length are simply computed as:

$$u_0 = -\frac{v_2}{v_1} \tag{17}$$

$$f = \sqrt{\frac{v_3}{v_1} - u_0^2} = \sqrt{\frac{v_3}{v_1} - \frac{v_2^2}{v_1^2}} = \sqrt{\frac{v_1 v_3 - v_2^2}{v_1^2}} \tag{18}$$

3.3 Extracting the Scaling Factor and the Extrinsic Parameters

Now that we have extracted the focal length and the principal point, we will show how the scaling factor s along with the extrinsic parameters (rotation matrix R and translation vector $\mathbf{t}$) can be computed using more constraints. Let us define a matrix A_i as:

$$\mathsf{A}_i = \lambda_i \begin{pmatrix} r_{11} & r_{12} & t_1 & 0 & 0 & 0 \\ s(r_{21}t_3 + r_{31}t_2) & s(r_{22}t_3 + r_{32}t_2) & st_2t_3 & sr_{21}r_{31} & sr_{22}r_{32} & s(r_{21}r_{32} + r_{22}r_{31}) \\ r_{31} & r_{32} & t_3 & 0 & 0 & 0 \end{pmatrix} \tag{19}$$

The subscript i refers to the ith scan of the calibration grid. We can first notice that:

$$\begin{aligned} t_{1i} &= a_{13}/\lambda_i \\ t_{2i} &= a_{23}/s a_{33} \\ t_{3i} &= a_{33}/\lambda_i \end{aligned} \tag{20}$$

It's easy to see that $\bar{\mathsf{R}}_i$ (the two first columns of R_i as defined in the previous subsection) can be expressed as:

$$\bar{\mathsf{R}}_i = \begin{pmatrix} \frac{1}{\lambda_i} & & \\ & \frac{1}{s a_{33i}} & \\ & & \frac{1}{\lambda_i} \end{pmatrix} \times \begin{pmatrix} a_{11i} & a_{12i} \\ a_{21i} - a_{31i}\frac{a_{23i}}{a_{33i}} & a_{22i} - a_{32i}\frac{a_{23i}}{a_{33i}} \\ a_{31i} & a_{32i} \end{pmatrix} \tag{21}$$

$$= \begin{pmatrix} \frac{1}{\lambda_i} & & \\ & \frac{1}{s} & \\ & & \frac{1}{\lambda_i} \end{pmatrix} B_i \tag{22}$$

where a_{xyi} are the elements of the matrix A_i and the matrix B_i defined as:

$$B_i = \begin{pmatrix} a_{11i} & a_{12i} \\ \frac{a_{21i}a_{33i} - a_{31i}a_{23i}}{a_{33i}^2} & \frac{a_{22i}a_{33i} - a_{32i}a_{23i}}{a_{33i}^2} \\ a_{31i} & a_{32i} \end{pmatrix} \tag{23}$$

As in the previous subsection, we once again make use of the orthogonality of the rotation matrix $\bar{\mathsf{R}}_i$ to gain constraints on λ_i and $\frac{1}{s t_{3i}}$. For instance, one notices that:

$$\bar{\mathsf{R}}_i^{\mathsf{T}} \bar{\mathsf{R}}_i = \mathsf{B}_i^{\mathsf{T}} \begin{pmatrix} \frac{1}{\lambda_i^2} & & \\ & \frac{1}{s^2} & \\ & & \frac{1}{\lambda_i^2} \end{pmatrix} \mathsf{B}_i = \mathtt{I}_{2\times 2} \tag{24}$$

The above result gives 3 linear equations in $\frac{1}{\lambda_i^2}$ and $\frac{1}{s^2}$. Since we solved for $\frac{1}{\lambda_i^2}$, the scaling factor λ_i is extracted up to a sign.

So far, only the 3 first columns of A_i have been used. In order to extract the real λ_i from the 2 possible solutions, the last 3 columns of A_i will be used. We proceed with the following simple steps for each possible solution:

- Compute $\bar{\mathsf{R}}_i$ and A_i from (21).
- Compute t_{1i} and t_{3i} as defined in (20).
- Compute the residual term:

$$\begin{aligned} \Delta = &(a_{24i} - s r_{21i} r_{31i})^2 + (a_{25i} - s r_{22i} r_{32i})^2 \\ &+ (a_{26i} - s(r_{21i} r_{32i} + r_{22i} r_{31i}))^2 \end{aligned} \tag{25}$$

The ideal solution is the one that leads to the smallest Δ. By the definition of a_{24i}, $a25i$ and a_{26i} (see (19)) and in an ideal noiseless case, Δ vanishes.

Notice that if a couple (s, λ_i) minimizes Δ, then $(-s, -\lambda_i)$ also minimizes Δ. This ambiguity corresponds to the mirror-pose solution. Given our choice of coordinate system, visible points must have positive Z-coordinate, thus we pick the solution that gives a positive t_{3i}.

Finally, t_{2i} is computed from (20) and the third column of the rotation matrix is obtained by a simple cross-product of the two columns of $\bar{\mathsf{R}}_i$. The orthonormality of the final rotation matrix R_i can be enforced using SVD.

Notice that, as opposed to the reference calibration method (Rajiv Gupta 1997), the proposed method estimates the scaling factor s related to the speed of the linear sensor.

3.4 Non-Linear Optimization

In this subsection, we give the details of a non-linear optimization procedure through bundle adjustment for our calibration method. Though optional, such optimization is highly recommended and as shown later, is fast and reduces the reprojection error.

Once an initial estimation of the internal parameters has been carried out (using the linear method described earlier), an optimization procedure can be applied in order to minimize the reprojection error in the camera and represented by following cost function:

$$\min_{\mathsf{K}, \bar{\mathsf{R}}_i, \mathbf{t}_i} \sum_{i,j} dist^2 \left((u_{ij}, v_{ij}, 1)^{\mathsf{T}}, \mathsf{K}\bar{\mathsf{R}}_i \mathbf{t}_i (a_j, b_j, 1)^{\mathsf{T}} \right) \tag{26}$$

where $(a_j, b_j, 1)^{\mathsf{T}}$ represents the jth feature on the calibration plane and $(u_{ij}, v_{ij}, 1)^{\mathsf{T}}$ its projection in the ith camera.

For each camera pose, we must optimize the 3 intrinsic parameters (supposed fixed) and the 6 extrinsic parameters (different at each pose). Thus, for n camera poses, we have $3 + 6n$ parameters to optimize. In our implementation, we used the Levenberg-Marquardt method for the optimization. Usual implementations take advantage of the sparsity of the problem to gain time on matrix operations such as inversions. However, given the small size of our problem, we used standard SVD routines to inverse matrices. Indeed, for a typical calibration process using 10 poses, solving for the normal equation involves inverting matrices of 63×63.

We give the formulation of the error function derivatives and the form of the jacobian matrix in the Appendix. Overall, the whole bundle adjustment process runs in less than 2 seconds on a 1.5 GHz laptop.

4 Complete Plane-Based Calibration Algorithm

In this section we present the complete plane-based algorithm for linear cameras calibration. From n scan of a calibration grid:

1. Estimate the projection matrices H_i for all n scans (see Sect. 3.1), using point matches and the relation

$$\begin{pmatrix} u_{ij} \\ v_{ij} \\ 1 \end{pmatrix} \sim \mathsf{H}_i \begin{pmatrix} a_j \\ b_j \\ 1 \\ a_j^2 \\ b_j^2 \\ a_j b_j \end{pmatrix} \tag{27}$$

where j is an index for calibration points. The estimation of H_i is equivalent to the so-called DLT (Direct Linear Transform) and can be done by solving a linear equation system.
2. Compute matrices M_i according to (16).
3. Form the matrix S of dimension $2n \times (3+n)$:

$$\mathsf{S} = \begin{pmatrix} m_{1,11}m_{1,12} & m_{1,11}m_{1,32} + m_{1,12}m_{1,31} & m_{1,31}m_{1,32} & m_{1,21}m_{1,22} & & \\ \vdots & \vdots & \vdots & & \ddots & \\ m_{n,11}m_{n,12} & m_{n,11}m_{n,32} + m_{n,12}m_{n,31} & m_{n,31}m_{n,32} & & & m_{n,21}m_{n,22} \\ M_{1,11}^2 - M_{1,12}^2 & 2(m_{1,11}m_{1,31} - m_{1,12}m_{1,32}) & M_{1,31}^2 - M_{1,32}^2 & M_{1,21}^2 - M_{1,22}^2 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ M_{n,11}^2 - M_{n,12}^2 & 2(m_{n,11}m_{n,31} - m_{n,12}m_{n,32}) & M_{n,31}^2 - M_{n,32}^2 & & & M_{n,21}^2 - M_{n,22}^2 \end{pmatrix} \tag{28}$$

4. Solve the following system to least squares:

$$\mathsf{S} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_{4,1} \\ v_{4,2} \\ \vdots \\ v_{4,n} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{29}$$

5. From the $v_{1,2,3}$, extract the intrinsic parameters f and u_0 according to (refeq.u0) and (18).
6. Compute s and the extrinsic parameters according to the algorithm of Sect. 3.3.
7. Optional but recommended: non-linear optimization of all unknowns, i.e. intrinsic and extrinsic parameters, by minimizing the reprojection errors (see Sect. 3.4).

5 Experimental Results

The proposed algorithm has been tested on both synthetic data and real data. Both tests are detailed in the next two subsections.

5.1 Computer Simulations

We performed several tests of our algorithm using synthetic data. Throughout all the experiments, we used a planar calibration grid of $10 \times 10 = 100$ corners. The virtual camera has a 1000×1000 image resolution, a focal length of 1000, and its principal point is located at the image center, at pixel (500, 500).

We refer to the "*calibration volume*" as the bounding box that encloses all the calibration grids. Actually the most relevant parameter is not the bounding box volume itself but its height. In our experiments, the volume height is expressed as a percentage of the grid's length. Some configuration examples with several calibration volumes are depicted in Fig. 2.

Sensitivity to Noise Level

For this test, we used 10 planes oriented randomly in a calibration volume of 100% the size of the calibration grid. After projection, a gaussian noise with mean 0 and increasing standard deviation was added to the image points. The standard deviation σ varied from 0.2 to 2. As in Zhang (1999), we performed 100 independent runs for each noise level and computed the average errors for both the focal length and the principle point. As we can see from Fig. 3 the error increases almost linearly for both the focal and the principal point. For an noise level of $\sigma = 0.5$ the errors in the focal and the principal point is less than 4 pixels which represents (given our camera characteristics) less than 0.8%.

Sensitivity to the Number of Planes

In this test, the sensitivity of our method w.r.t the number of planes is investigated. We set the calibration volume height

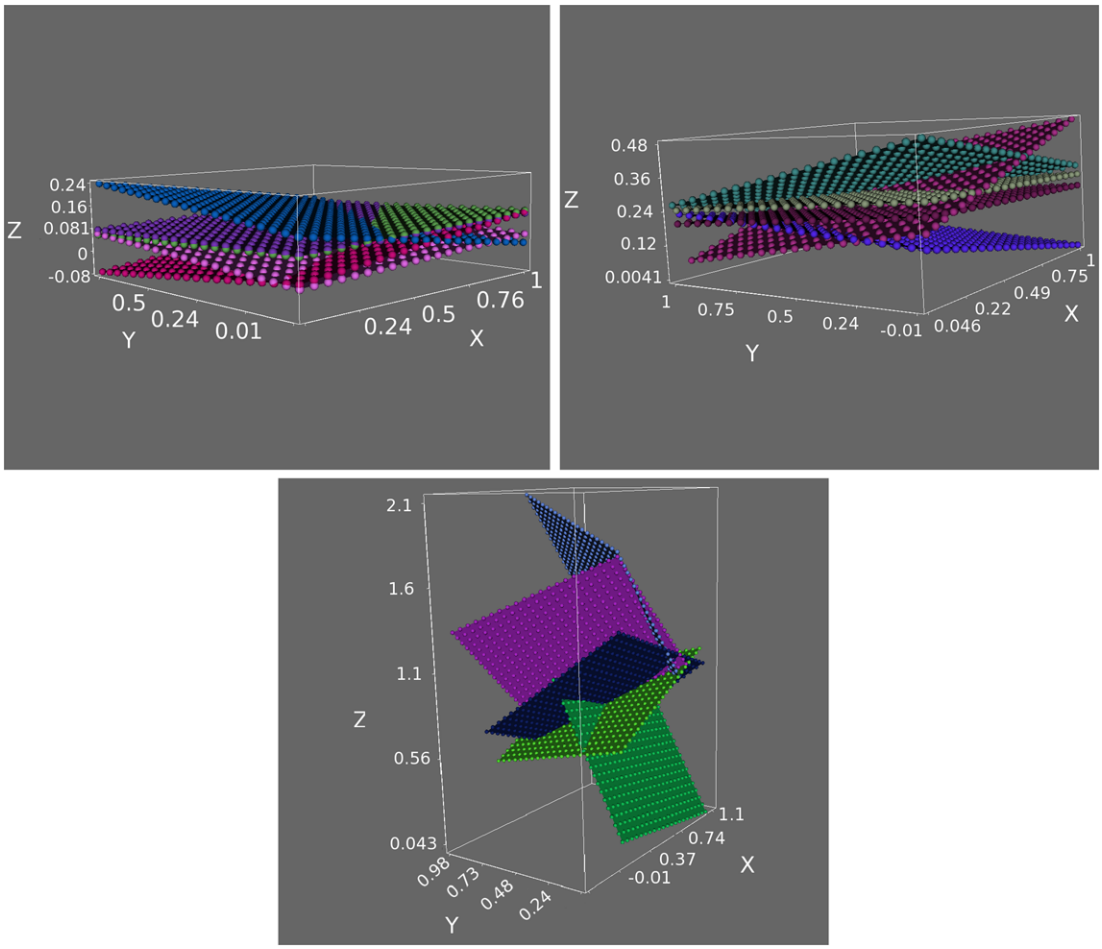

Fig. 2 An example of 3 calibration volume with increasing height. From *left* to *right*, 25%, 50% and 200% of the calibration length

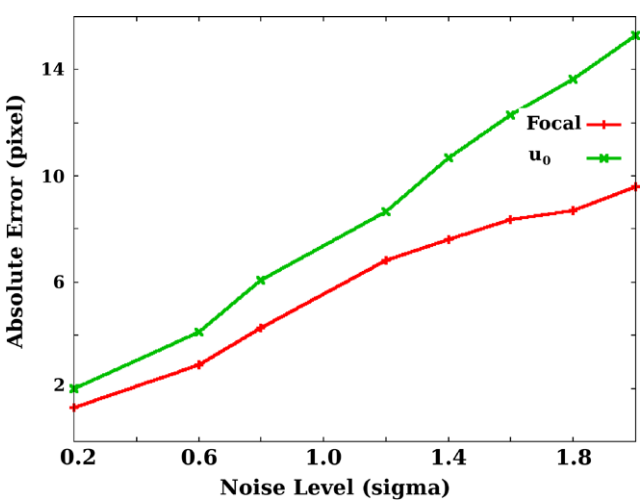

Fig. 3 Focal length and principal point errors w.r.t the noise level in the image points

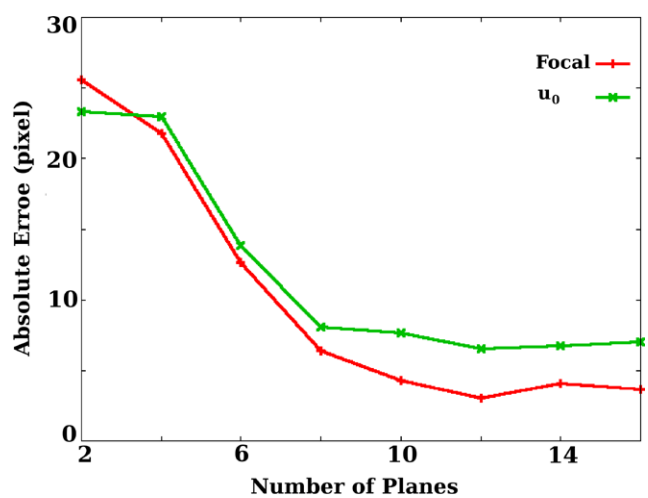

Fig. 4 Focal length and principal point errors vs. the number of planes used ($\sigma = 0.5$)

to 100% of the grid's length and we varied the number of planes from 2 to 20. The average errors (from 100 independent runs) for both the focal length and the principal point were estimated and reported on Fig. 4 for a noise level of $\sigma = 0.5$ and $\sigma = 1.0$. We notice that the errors decrease as more planes are used.

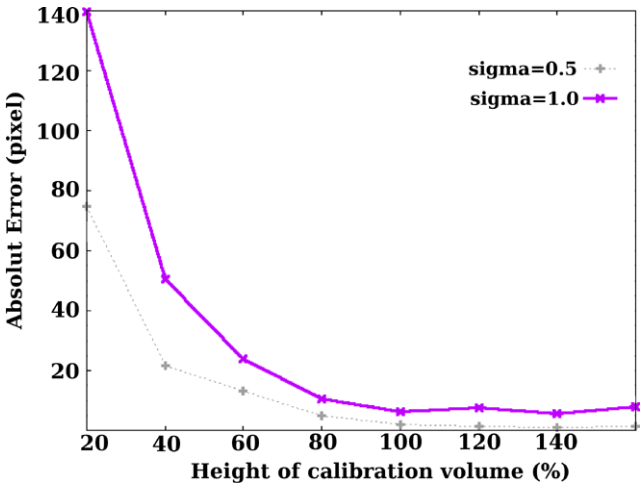

Fig. 5 Focal length error vs. the height of calibration volume

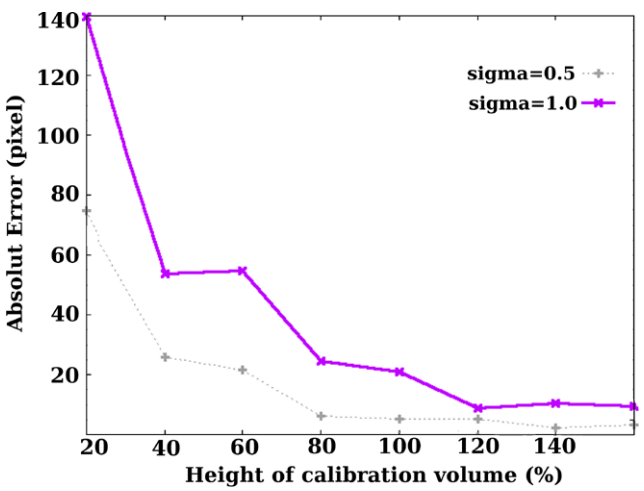

Fig. 6 Principal point error vs. the height of calibration volume

Sensitivity w.r.t the Reconstruction Volume

In this last synthetic experiment we analyse the performance of our method with respect to the calibration volume, or more precisely the volume's height. For this test we used 10 calibration grids oriented randomly and varied the calibration volume height from 20% to 160% of the grid's length (we remind that the grid is squared). This test was performed with a noise level of $\sigma = 0.5$ and $\sigma = 1.0$ (which is larger than the noise observed in a typical calibration (Zhang 1999)). We can see from Figs. 5 and 6 that the volume's height affects the quality of the calibration. In fact the errors decrease when a higher reconstruction volume is used. This is primarily due to the fact that a higher reconstruction volume permits a higher motion degree which guarantees a better sampling of the rotation space.

Table 1 Results of the camera calibration as Push-Broom and fully perspective (see text)

Parameter	Perspective	Push-Broom	Error (%)
Focal length	1983.98	1998.32	0.7
Principal point	554.81	549.68	0.9
Scale factor	31.75	32.56	2.4

5.2 Real Data

Experiments on real data were conducted on two setups. The first one consists of a regular perspective camera mounted on a linear stage to simulate a pushbroom camera. In the second experiment we will show how a consumer flatbed scanner can be modeled as a pushbroom sensor. Because flatbed scanners are widely available and very affordable, they make a perfect device for high resolution measurements.

Camera + Linear Stage

For this experiment, we mounted a Prosilica camera on a controllable linear stage. The camera was set to deliver images of 1360×1024 pixels at 5 frame per second. The speed of the stage was set to 4 mm/s. The size of the squares on the calibration plane were 1×1 inch.

From each image delivered by the camera, we extracted the column that passes by the principal point and form a *panorama* by stacking them on top of each other. Hence, the resulting panorama is akin to an image shot with a pushbroom camera (Seitz and Kim 2002). This procedure was repeated to acquire 10 images of the calibration plane under several orientations. The results of our calibration are shown in Table 1. To assess the quality of our calibration, we also included the intrinsic parameters of the camera when calibrated as fully perspective. The later has been performed using the *OpenCV* library plane-based calibration routines. The results of this experiment are reported in Table 1.

We can see that the estimated perspective parameters (focal and principal point) are compatible with the results obtained using a standard plane-based calibration. The scale factor in the perspective column of Table 1 is the expected value given our settings and is computed as follow. Within one second the camera acquires 5 frames, thus 5 columns of the push-broom image. In this same second the camera would have translated by 4 mm $= \frac{4}{25.4}$ in. Since one unit of the calibration grid is 1 inch, the scale factor is $\frac{1\times 5}{\frac{4}{25.4}} = 31.75$. Our method estimated a scale factor of 32.56, yielding an error of 2.4%. The results of our algorithm include the non-linear refinement stage. However, we have noticed a negligible improvement in the error function when compared to the original linear solution.

Fig. 7 Our setup to simulate a pushbroom camera. The camera (Prosilica) is mounted on a programmable linear stage. The accuracy of the stage is in the 100th of millimeter

Table 2 Flatbed scanner calibration results (see text)

Parameter	DLT	Plane-based	Error (%)
Focal length	2673.4	2659.7	0.57
Principal point	1315.2	1299.5	1.2
Scale factor	–	146.48	2.4

Flatbed Scanner

We tested the proposed algorithm on an Epson V200 flatbed scanner. The manufacturer claims that the scanner is suited for scanning 3D objects thanks to its depth of field and adapted optic. We thus, modeled the scanner as a pushbroom camera and used the proposed algorithm to retrieve its intrinsic parameters using a planar grid. The scans were done at a resolution of 300 dpi (dot per inch), the grid's squares were half inch long each. Resulting images had a resolution of 2538×2328. Homographies were estimated by first detecting grid's features using *OpenCV* routines. To ensure a better numerical stability, points were normalized as suggested in Hartley (1997). We also calibrated the same scanner using the DLT method proposed by Hartley (Rajiv Gupta 1997; Gupta and Hartley 1995). In the later case, we scanned a 3D calibration rig and features were manually selected. Results and comparisons are reported in Table 2.

Since no ground truth was available, we took as a reference the classical calibration method proposed by Hartley et al. (Rajiv Gupta 1997; Gupta and Hartley 1995) and we can see that the focal length and the principle point estimated by our method are very close to the estimation made by Hartley's method (both parameters differ by less than 1.5%). Further, each square of the calibration grid measured 0.5 inch length and giving the fact that the tests were made at a resolution of 300 dpi, the scaling factor s should be $s = 300 \times 0.5 = 150$ which differs by only 2.4% from the scaling factor computed using our method.

6 Conclusion

In this paper we have presented a simple algorithm to calibrate a linear camera. The calibration is done using images of a planar grid acquired under different orientations. The proposed method is based on a closed-form solution with an optional non-linear refinement. Both synthetic and real experiments proved the effectiveness and the quality of our procedure. As opposed to the reference method, the proposed one estimates all three internal parameters including the scaling factor, and the calibration tool is as simple as a planar grid.

Appendix: Bundle Adjustment

In this appendix we give the details of our bundle adjustment implementation. The emphasis is given to the partial derivatives formulation to estimate the jacobian. Details of the bundle adjustment algorithm itself can be found in Hartley and Zisserman (2003) and Triggs et al. (2000).

A.1 Parametrization

The entries of the intrinsic matrix are parameterized by the focal length f, the principal point u_0 and the scale factor s. For each pose i, the translation vector t_i is represented by its entries (t_{1i}, t_{2i}, t_{3i}) and the rotation R_i is parameterized by computing, at each iteration, update rotations with small angles Δ_i, relative to the rotations of the previous iteration S_i:

$$\mathsf{R}_i = \Delta_i \mathsf{S}_i$$

The matrix Δ_i is the skew-symmetric matrix that encodes the cross product with the vector (w_{1i}, w_{2i}, w_{3i}). Its direction represents the axis of the rotation update and its norm represents the update rotation angle.

A.2 Partial Derivatives

We define the residual e of the cost function we wish to minimize (see (26)) in terms of its components $e1$ and $e2$ as follows:

$$e1 = u_{ij} - \frac{(\mathsf{K}[\bar{\mathsf{R}}_i(a_j, b_j, 1)^\mathsf{T} - \mathbf{t}_i])_1}{(\mathsf{K}[\bar{\mathsf{R}}_i(a_j, b_j, 1)^\mathsf{T} - \mathbf{t}_i])_3}$$

$$e2 = v_{ij} - (\mathsf{K}[\bar{\mathsf{R}}_i(a_j, b_j, 1)^\mathsf{T} - \mathbf{t}_i])_2$$

We recall that $(u_{ij}, v_{ij})^{\mathsf{T}}$ is the projection of the jth grid point in the camera i. The derivatives of the residual error w.r.t the intrinsic parameters are:

$$\begin{pmatrix} \frac{\partial e1}{\partial f, u_0, s} \\ \frac{\partial e2}{\partial f, u_0, s} \end{pmatrix} = \begin{pmatrix} \frac{-\lambda_1}{\lambda_2} & -1 & 0 \\ 0 & 0 & -t_3 - \lambda_3 \end{pmatrix}$$

Notice that the derivatives are obtained after setting the values for the update angles (w_1, w_2, w_3) at zero. The derivatives w.r.t the translations are:

$$\begin{pmatrix} \frac{\partial e1}{\partial t_1, t_2, t_3} \\ \frac{\partial e2}{\partial t_1, t_2, t_3} \end{pmatrix} = \begin{pmatrix} -\frac{f}{\lambda_2} & 0 & \frac{f\lambda_1 + u_0\lambda_2}{\lambda_2^2} - \frac{u_0}{\lambda_2} \\ 0 & -s & 0 \end{pmatrix}$$

The derivatives of the residuals w.r.t the rotation updates are:

$$\begin{pmatrix} \frac{\partial e1}{\partial w_1, w_2, w_3} \\ \frac{\partial e2}{\partial w_1, w_2, w_3} \end{pmatrix} = \begin{pmatrix} \frac{u_0\lambda_3}{\lambda_2} + \frac{-\lambda_3\lambda_1 f - \lambda_3\lambda_2}{\lambda_2^2} & \frac{\lambda_1\lambda_4 f + \lambda_4\lambda_2 u_0}{\lambda_2^2} - \frac{\lambda_2 f - t_3 f + \lambda_4 u_0}{\lambda_2} & \frac{\lambda_3 f}{\lambda_2} \\ -s(t_3 - \lambda_2) & 0 & -s(\lambda_1 - t_1) \end{pmatrix}$$

where the intermediate variables $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are defined as follows:

$$\lambda_1 = ar_{11} + br_{12} + t_1$$

$$\lambda_2 = ar_{31} + br_{32} + t_3$$

$$\lambda_3 = ar_{21} - br_{22}$$

$$\lambda_4 = ar_{11} - br_{12}$$

A.3 The Jacobian

Using the partial derivatives of the residual, the formulation of the jacobian $\mathbf{J}$ of the cost function defined in (26) is straightforward. We give the example of the jacobian for $i = 3$ camera poses:

$$\mathbf{J} = \left(\begin{array}{c:c:c:c} \frac{\partial e}{\partial f, u_0, s} & \frac{\partial e}{\partial t_{11}, t_{21}, t_{31}, w_{11}, w_{21}, w_{31}} & 0 & 0 \\ \hdashline \frac{\partial e}{\partial f, u_0, s} & 0 & \frac{\partial e}{\partial t_{12}, t_{22}, t_{32}, w_{12}, w_{22}, w_{32}} & 0 \\ \hdashline \frac{\partial e}{\partial f, u_0, s} & 0 & 0 & \frac{\partial e}{\partial t_{13}, t_{23}, t_{33}, w_{13}, w_{23}, w_{33}} \end{array}\right)$$

References

Basler (2009). Basler vision technologies. http://www.baslerweb.com/.

Gupta, R., & Hartley, R. I. (1995). Camera estimation for orbiting pushbrooms. In *The proc. of second Asian conference on computer vision* (Vol. 3).

Gurdjos, P., & Payrissat, R. (2001). Calibration of a moving camera using a planar pattern: a centre line-based approach for varying focal length. In *British machine vision conference, session 7: geometry & structure*.

Hirschmüller, H. F. S., & Hirzinger, G. (2005). Stereo vision based reconstruction of huge urban areas from an airborne pushbroom camera (hrsc). In W. G. Kropatsch, R. Sablatnig, & A. Hanbury (Eds.), *Lecture notes in computer science: Vol. 3663. DAGM-symposium* (pp. 58–66). Berlin: Springer.

Ishiguro, H. Y. H., & Tsuji, S. (1992). Omni-directional stereo. *IEEE, Transactions on Pattern Analysis and Machine Intelligence*, 257–262.

Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(6), 580–593.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press. doi:10.2277/0511188951.

Faugeras, O. G. T. (1987). Camera calibration for 3d computer vision. In *Int. workshop on machine vision and machine intelligence* (pp. 240–247).

Rajiv Gupta, R. I. H. (1997). Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(9), 963–975.

Ricolfe Viala, C., & Sanchez Salmeron, A. (2005). Improving accuracy and confidence interval of camera parameters estimated with a planar pattern. In *International conference on image processing* (Vol. II, pp. 1142–1145).

Seitz, S. M., & Kim, J. (2002). The space of all stereo images. *International Journal of Computer Vision*, *48*(1), 21–38.

Shum, H. Y., & Szeliski, R. (1999). Stereo reconstruction from multiperspective panoramas. In *International conference on computer vision* (pp. 14–21).

Sturm, P., & Maybank, S. (1999). On plane-based camera calibration: a general algorithm, singularities, applications. In *Proc. IEEE conf. on computer vision and pattern recognition* (Vol. I, pp. 432–437).

Triggs, B., Mclauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). *Bundle adjustment—a modern synthesis* (Vol. 1883). http://www.metapress.com/link.asp?id=PLVCRQ5BX753A2TN.

Tsai, R. (1986). An efficient and accurate camera calibration technique for 3-d machine vision. In *Proc. IEEE conf. on computer vision and pattern recognition* (pp. 364–374).

Wang, J., Shi, F., Zhang, J., & Liu, Y. (2006). Camera calibration from a single frame of planar pattern. In *Advanced concepts for intelligent vision systems* (pp. 576–587).

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *International conference on computer vision* (pp. 666–673).